

REPAST PARAMETER SWEEPS GETTING STARTED

MARK BRAGEN & MARK ALTAWEEL

1. PARAMETER SWEEPS

The Repast Simphony User Interface has been extended to provide a mechanism for creating parameter sweep and optimized sweep definition files. These definition files can then be submitted for execution in either a distributed grid environment or on the local computer. Fig. 1 is a screen capture of the Parameter Sweep window for the Predator Prey Model.

The window is divided into four sections:

- (A) Specify the repeat count for the sweep
- (B) Specify the parameter value definitions for a single sweep
- (C) Specify the require file information based on execution mode
- (D) Run submission

In Section A (see Fig. 2), the user specifies the number of times that the parameter sweep definitions from Section B will be executed. Note that the parameter sweep definition in Section B specifies a single unit of work that will be executed (i.e. multiple model executions with varying parameter values).

In Section B (Fig. 3), the user specifies the values for each of the model parameters. Except for the Default Random Seed value, the value(s) can be specified in one of three ways:

- (1) A constant value
- (2) A list of blank separated values
- (3) A list of values specified with from, to, step definition

These are selected from the drop-down selection. Values must be specified for all model parameters. The Default Random Seed has an additional option of a completely random specification. The Random Seed will be determined during the execution of the model.

In Section C, the user specifies the type of execution. Three types, each of which has different information requirements, are available:

- (1) Local Execution Execution of the model on the local machine

- (2) Optimized Execution Execution of the model on the local machine using a specified optimizer
- (3) Grid Execution Execution of the model across GridGain nodes.

In Local Execution mode, the user simply needs to specify the Scenario Directory (see Fig. 4). Note that this text field is filled by using the scenario directory provided as an argument to the launcher.

In Optimized Execution mode, the user must specify the Run Result Producer and the Advancement Chooser in addition to the Scenario Directory (see Fig. 5). Please reference the definitions of these two items in the batch processing documentation.

In order to use Grid Execution mode, the user must download and install the GridGain software package and complete the GridGain and Repast Simphony Grid Extension installation procedure as documented below. If GridGain is installed, the Model Execution portion of the window will appear as in Fig. 6 where the Grid Execution radio button and the two check boxes are enabled. If GridGain is not installed, the Model Execution portion of the window will appear as in Fig. 7 where the radio button and check boxes are disabled and the GridGain installation requirement is noted.

In addition to the Scenario Directory, the user must specify: the Project Directory, the batch XML file, the Remote Project Directory, and the remote Batch XML file (see Fig. 6). These items are defined in Section 2. Note that there are two checkboxes activated in Grid Execution mode: Retain Remote Files and Create Jar. The Retain Remote Files checkbox specifies whether or not the model output files will be deleted on the remote GridGain nodes. The Create Jar checkbox specifies whether or not the project jar file will be created for the user. The Save Configuration button can be used to save in the scenario directory a file containing the definitions for all three execution modes that will be loaded during future executions of the scenario.

Finally, Section D (see Fig. 8) contains the button that will submit the job(s) for execution in one of the three modes. Note that the button text will reflect the execution mode that has been selected in Section C.

When the parameter sweep job has been submitted for execution, a console log window will open and display messages from Repast Simphony and/or GridGain depending on the execution type. See Fig. 9 for an example of this window.

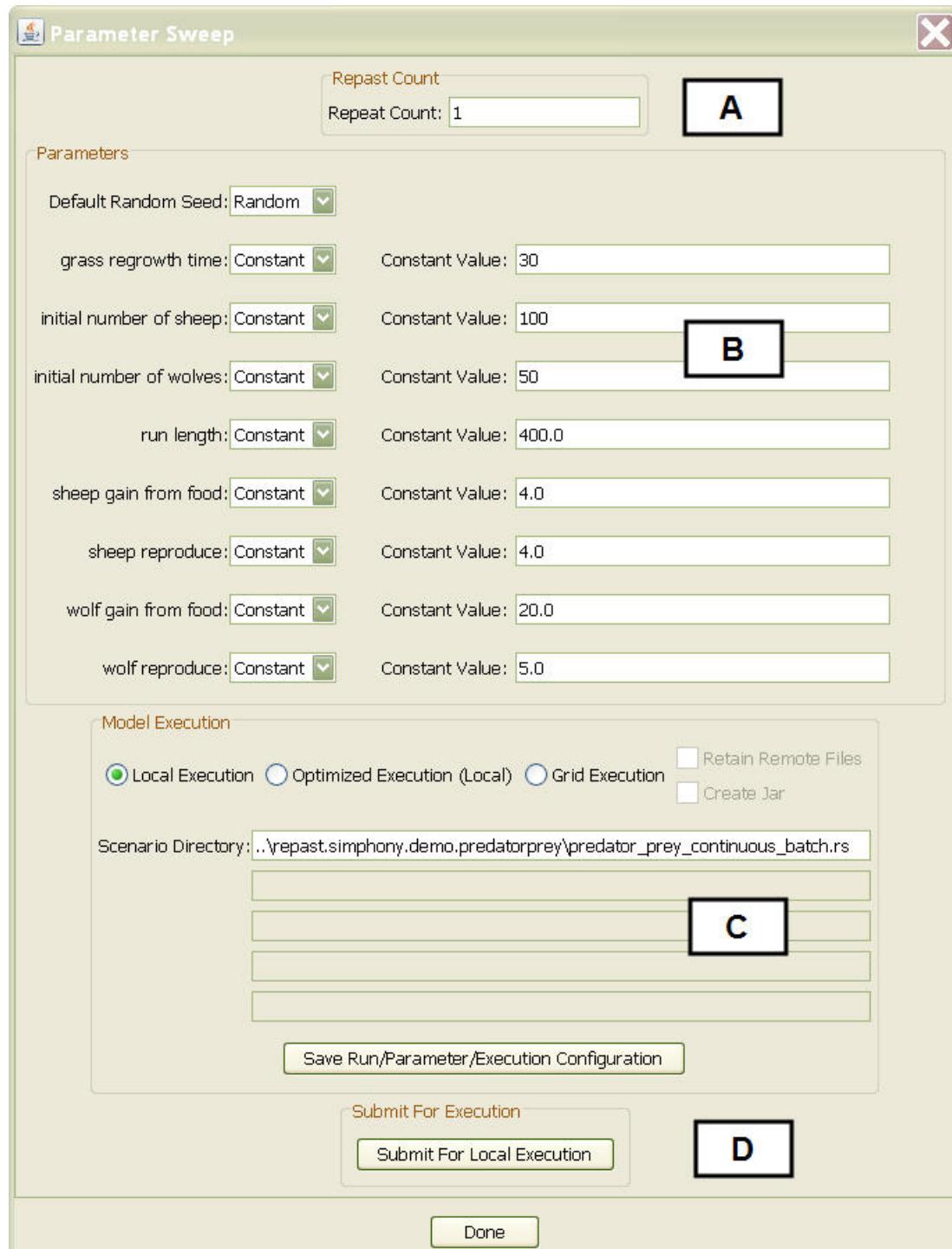


FIGURE 1

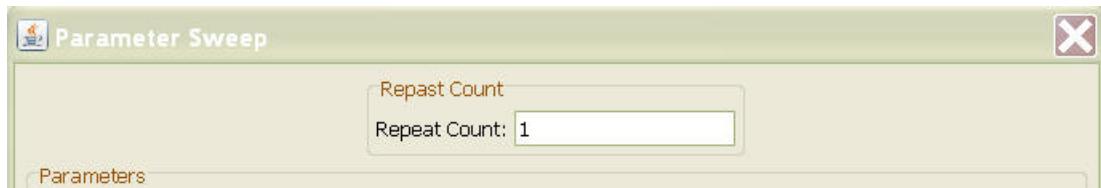


FIGURE 2

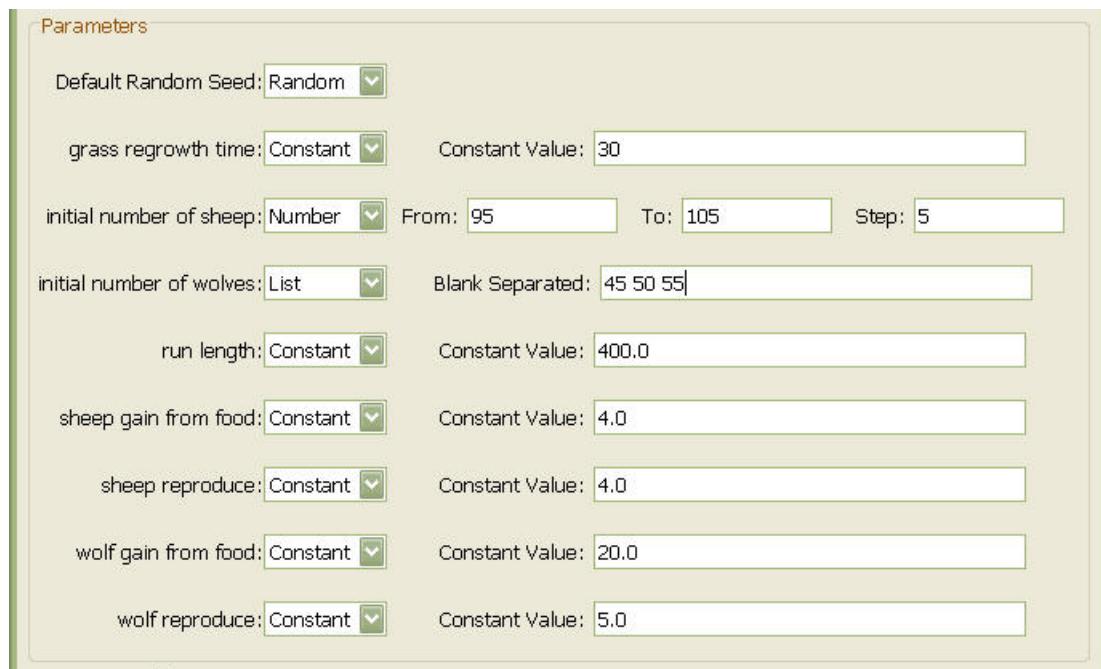


FIGURE 3

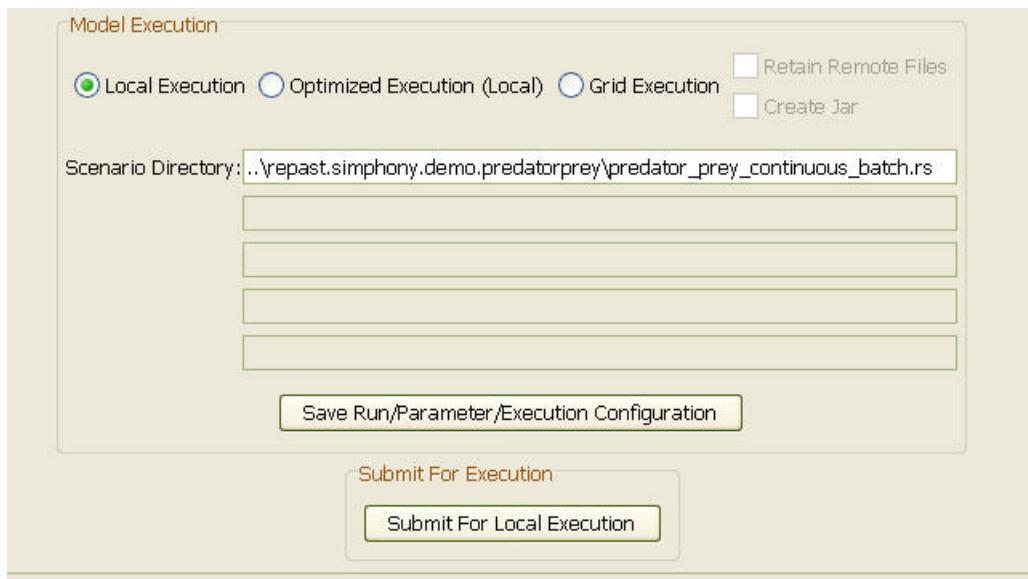


FIGURE 4

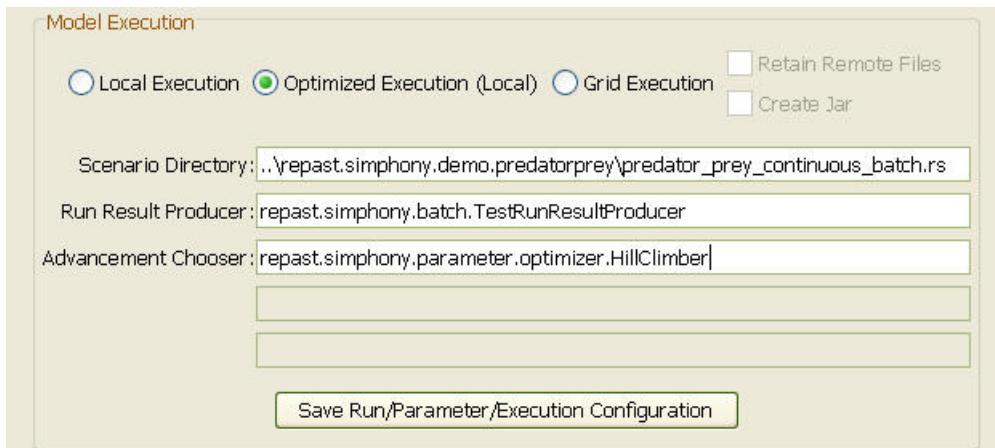


FIGURE 5

Model Execution

Local Execution Optimized Execution (Local) Grid Execution Retain Remote Files
 Create Jar

Scenario Directory: ..\repast.simphony.demo.predatorprey\predator_prey_continuous_batch.rs

Project Directory: ..\repast.simphony.demo.predatorprey

Batch XML: ..\repast.simphony.demo.predatorprey\predatorprey_batch\batch_params.xml

Remote Scenario Directory: c:\Util\gridgain-2.1.1\bin\predator_prey_continuous_batch.rs

Remote Batch XML: c:\Util\gridgain-2.1.1\bin\predator_prey_batch\batch_params.xml

FIGURE 6



FIGURE 7

Model Execution (NOTE: Grid Execution Requires GridGain Installation)

Local Execution Optimized Execution (Local) Grid Execution Retain Remote Files
 Create Jar

Scenario Directory: ..\repast.simphony.demo.predatorprey\predator_prey_continuous_batch.rs

FIGURE 8

FIGURE 9

2. DEVELOPING A DISTRIBUTED BATCH REPAST SIMPHONY PROJECT

A new feature within Repast Simphony is a largely automated distributed batch framework that allows users to distributed their simulations in batch mode using multiple computer nodes and/or cores. This feature takes advantage of Repast Simphony's previous batch development but adds a new layer of capability in a plugin called repast.simphony.distributedBatch. The instructions below detail how users can setup their own distributed batch process using a Predator Prey project example. Previous Repast capability in allowing users to conduct single node batch runs are still enabled in the 2.0 version of Repast Simphony. This is done, as in previous versions of Repast Simphony, in the repast.simphony.batch plugin.

2.1. Downloading and Installing GridGain. The first step in using distributed batch for your project is to go to the GridGain website (<http://www.gridgain.com/>) and download GridGain 2.1.1 at:http://www.gridgain.com/past_downloads.html or at the following address <http://mac.softpedia.com/get/Development/Java/Gridgain.shtml>. Repast Simphony is currently using GridGain 2.1.1 because during the time of development this was the stable version released by GridGain. In addition, a new licensing scheme by GridGain makes the 2.1.1. version more feasible for Repast Simphony than the newer GridGain 3.0 version. For now, the GridGain 2.1.1 version should work well on Windows (Windows 2000 and up), Mac (Mac OS X versions), and Linux systems. Users will need to install GridGain 2.1.1 on all nodes, including the computer that they are using to launch Repast Simphony simulations (Fig. 10). Users should read the GridGain 2.1.1 installation instructions for proper configuration of GridGain 2.1.1. Once installed, place the repast.simphony.batch.jar, which is found in the /transferFiles folder of the repast.simphony.distributedBatch plugin, into the /libs/ext folder located within the GridGain installation folder (i.e., [location of the GridGain 2.1.1 folder]/libs/ext) for all nodes that have GridGain installed and those that will be used for distributed Repast batch runs.

The default-spring.xml file, found in repast.simphony.distributedBatch, should be used to replace the default-spring.xml file in the /config folder within the GridGain 2.1.1 installation folder on all nodes used (Fig. 11). This file will allow the default distributed configuration on all nodes used to be the same as that found in the repast.simphony.distributedBatch folder, which controls the configuration of the user's computer where processes are launched from.

The next step is creating a standard batch file, often called batch_params.xml, which contains the simulation parameters and number of batch runs. The documentation above described how to create one using the batch parameters GUI. You can do that or create one by hand. Because the project is a distributed batch project, users should set the "sweep runs" setting to 1, as the sweep setting will now be handled by the distributed process. If a user desires to run a regular batch run (i.e., using only one node), then the user should set the sweep runs setting to whatever number he or she desires. (Fig. 12) below shows what a standard batch_params.xml file looks like. This file should be placed somewhere within the user's Repast project folder. You should also remember to setup outputs as you normally would for Repast Simphony projects (e.g., text file outputters).



FIGURE 10. The GirdGain 2.1.1 installer for Mac OS X.

2.2. Creating the Project Jar File. The user then has two options for creating a project jar file to be used in the distributed runs. The first option is to use the GUI described above, and make sure the Create Jar options is selected.

Alternatively, the Eclipse Repast Simphony IDE can be used. In this option, the user needs to click on their project and create a jar file by choosing File → Export→ JAR file. Then, in the “Jar Export” window (Fig. 13), name the project jar and export it. The user will then need to place this jar file in the /transferFiles folder located in their project folder.

2.3. Setting the Distributed Batch Runs. Next, the user sets up a project that uses the repast.simphony.distributedBatch plugin. There are several path settings that need to be determined for both the local node (i.e., the computer you are using to launch the process) and the remote nodes. First, look at the XML_Launch_Inputs.xml file located in the /launchData folder within the repast.simphony.distributedBatch plugin. You will need to edit the following sets of paths and data (Fig. 14 & Fig. 15).

Repast Simphony applies a “user_path.xml” file that is typically located in the user’s .rs folder (see repast.simphony.demo.predatorprey as an example). This file allows Repast Simphony to loaded necessary classes for simulation runs and is used for regular GUI simulations, regular batch simulation, and distributed batch runs. In addition to setting the local bin folder’s location, the user needs to indicate where the remote nodes’ class bin

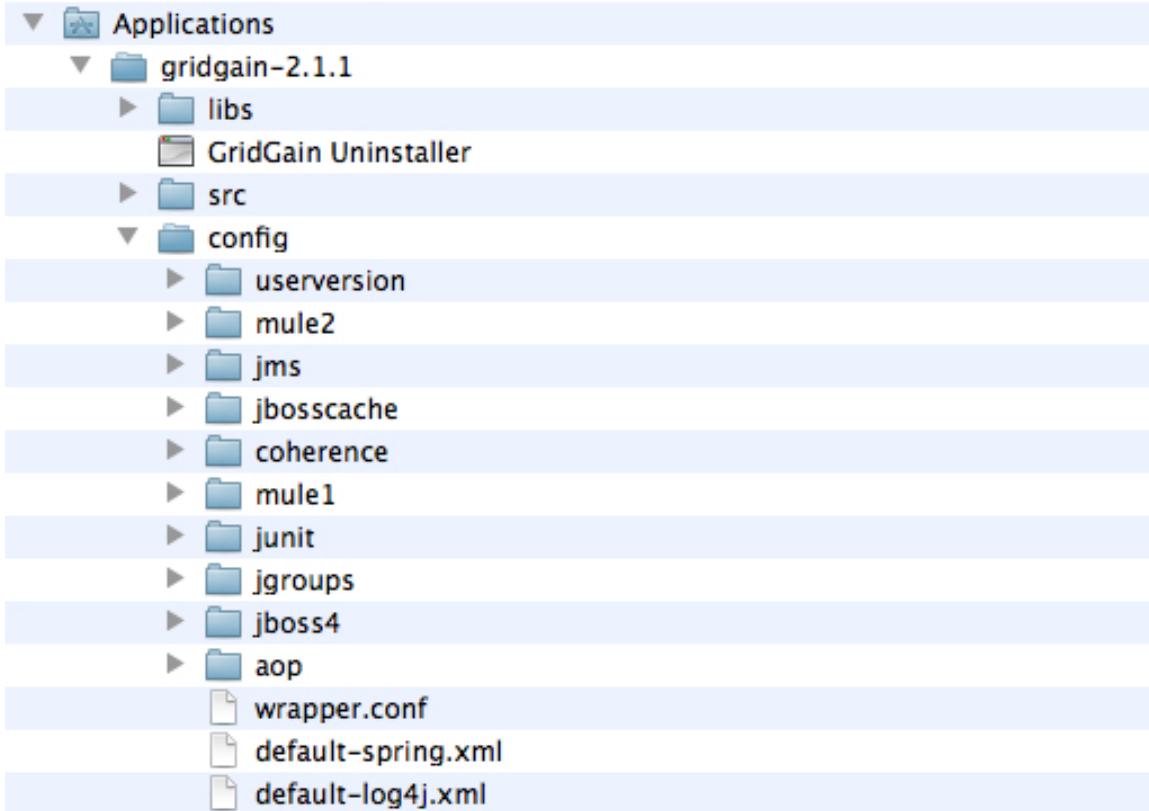


FIGURE 11. default-spring.xml file location

folder will be located. In the repast.simphony.distributedBatch plugin, the processes will launch on the remote nodes from GridGain's installation /bin folder. When a distributed processes is launched, the user's .jar project file will be unjared, exposing the class files in the jar file. The class files will be typically located in the GridGain /bin folder at that point. Thus, users will need to indicate a path where the users' classes are located in GridGain. Typically, the user should place a class reference to the GridGain bin folder (Fig. 16). However, if the jar was created the gui they may need to reference their project's bin folder within the bin of GridGain (Fig. 17). The two /bin references are needed because the user's jar has its own bin folder.

2.4. Launching the Remote Nodes. The user is now ready to launch the remote nodes. First, be sure to have the JAVA_HOME and GRIDGAIN_HOME variables set on the remote nodes, as stated by the GridGain installation instructions (please read the GridGain 2.1.1 instructions prior to using GridGain in Repast). To launch GridGain, simply login into

```

1<?xml version="1.0"?>
2<sweep runs="1">
3
4  <parameter name="initialnumberofwolves" type="constant" constant_type="number" value="50"/>
5  <parameter name="initialnumberofsheep" type="constant" constant_type="number" value="100"/>
6  <parameter name="wolfgainfromfood" type="constant" constant_type="number" value="20.01"/>
7  <parameter name="sheepgainfromfood" type="constant" constant_type="number" value="4.01"/>
8  <parameter name="wolfreproduce" type="constant" constant_type="number" value="5.01"/>
9  <parameter name="sheepreproduce" type="constant" constant_type="number" value="4.01"/>
10
11  <parameter name="grassregrowthtime" type="constant" constant_type="number" value="30"/>
12
13  <parameter name="runlength" type="constant" constant_type="number" value="400.01"/>
14</sweep>

```

FIGURE 12. Example Batch File

your remote nodes and launch the “gridgain.sh” file or “gridgain.bat” file (.sh for Mac and Linux and .bat for Windows machines). For Mac and Linux machines, type “sh [path to gridgain.sh file]/gridgain.sh” in the line command on the remote node, which should then activate your remote node. For Windows machines, just type the following path “[path to gridgain.bat file]/gridgain.bat” to do the same process as the other operating systems. Once launched, you do not need to turn off any nodes if you plan on running multiple sets of distributed batch runs. However, if any launch errors occur on the remote nodes or you edit your code, then you will need to restart the remote nodes as the new classes or updated classes will need to be reloaded on the remote machines. You may also want to configure a startup script that will automatically launch your GridGain startup files on the remote nodes. Once launched, you should see the remote nodes producing log information that indicates they are ready to receive processes launched from your local computer and can connect to other nodes in a cluster (Fig. 18).

2.5. Launching the Distributed Process. The user can now launch the distributed process. The user can now revert to using the GUI Parameter Sweep setup (see above) or use a Java Application launch configuration in Eclipse. To do this second option, go to Run Configurations... in the Repast Simphony Eclipse IDE and in the run configurations window select the Java Application setup for BatchMainSetup. There, choose the Classpath tab in the Run Configurations window. Add the local user project’s bin folder to the User Entries (Fig. 19). Then, in the Environment tab edit the local path to the GRIDGAIN_HOME directory of the GridGain installation folder (Figure 11). This will automatically startup your local GridGain application at runtime. Users can also create their

own Run Configurations... setup using similar settings to the "BatchMainSetup.launch" file located in the Repast Simphony distributedBatch plugin folder.

Now, the user can select the Run button in the Run Configurations window (bottom right corner in Run Configurations) and launch the distributed process. The log should produce output information in the local console window (Fig. 21) and remote nodes (Fig. 22), showing the distributed process running. The output should include information on the number of nodes running in GridGain and any output console information produced by the user's project (e.g., in the Predator Prey example the outputter, setup in the Repast GUI environment, displays user output). You should also see unique threads running on the local and remote nodes. The example Predator Prey project ((Fig. 21 and Fig. 22) shows two nodes (i.e., two computers) executing the model. Another feature in distributed batch is that it uses automatic failover, which allows other nodes to take the jobs from machines that may fail during the batch process.

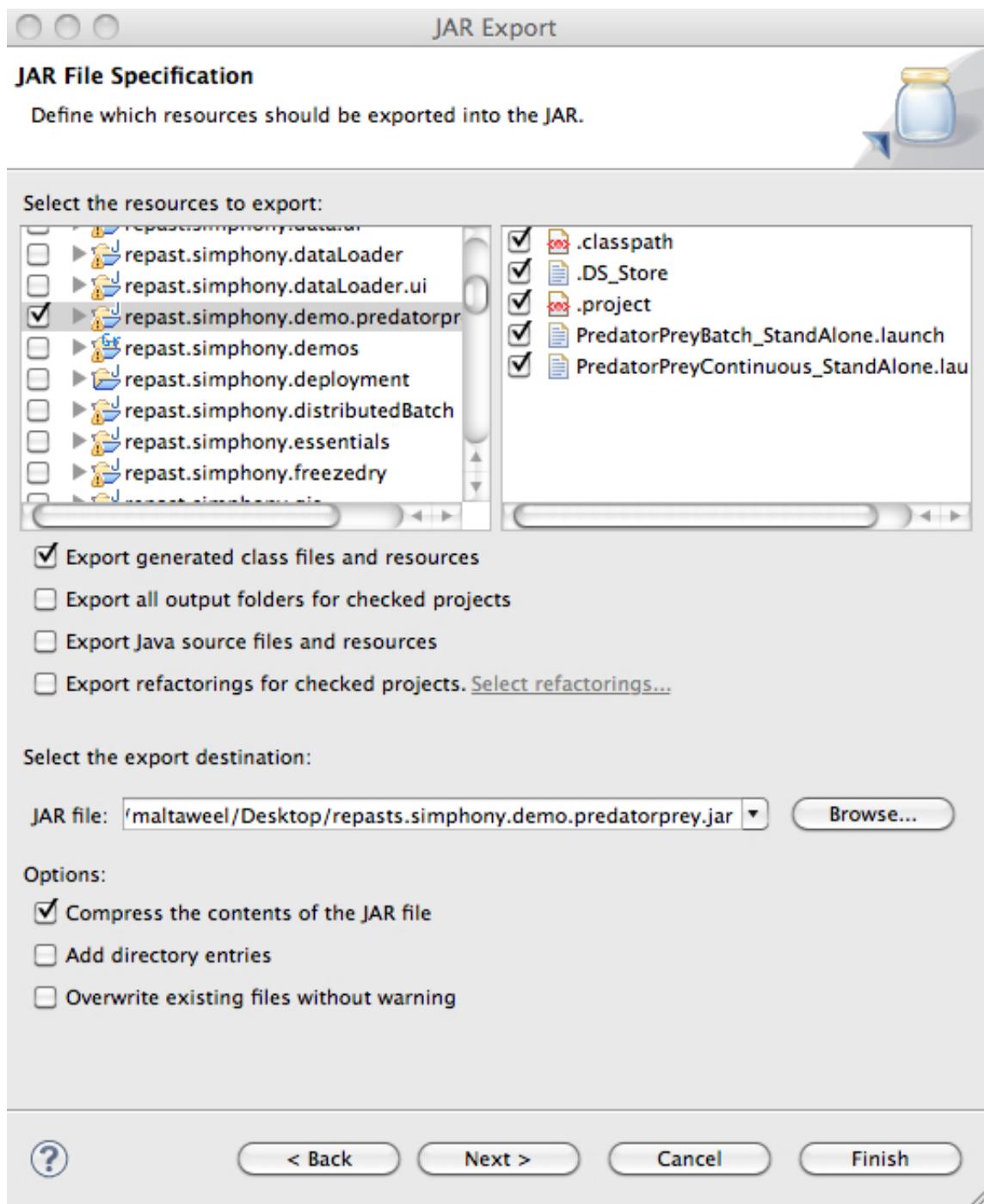


FIGURE 13. Jar Exporting

Name of Variable	Significance
numberOfRun	Number of distributed runs
name	Name of the distributed project user folder
mainScenario	Local node's location of the .rs folder
projectPath	Path to the project on the local node
batchXMLPath	path to local node's batch file
remoteBatchXML	path to remote node's batch file
scenario	path to remote node's .rs scenario folder
delayTime	time (in milliseconds) to delay between batch runs to allow automated retrieval of remote output files (optional and can be 0)
removeFiles	boolean to remove or not remove remote output files on remote nodes after simulation runs
createJar	flag to automatically create a user jar for distribution at runtime using build.xml

FIGURE 14. XML_Launch_Inputs.xml file's user inputs

```
<?xml version="1.0" encoding="UTF-8"?>
<XMLInputData>
<numberOfRuns>4</numberOfRuns> <!-- how many runs -->
<name>repast.simphony.demo.predatorprey</name> <!-- name of the user project folder -->
<mainScenario>/Applications/RepastUserEnvironment11/workspace/repast.simphony.demo.predatorprey/predator_prey_continuous_batch.rs</mainScenario>
<projectPath>/Applications/RepastUserEnvironment11/workspace/repast.simphony.demo.predatorprey</projectPath> <!-- path to the project on the local node -->
<batchXMLPath>/Applications/RepastUserEnvironment11/workspace/repast.simphony.demo.predatorprey/predator_prey_batch/batch_params.xml</batchXMLPath>
<remoteBatchXML>/Applications/gridgain-2.1.1/bin/predator_prey_batch/batch_params.xml</remoteBatchXML> <!-- path to the remote batch XML file -->
<scenario>/Applications/gridgain-2.1.1/bin/predator_prey_continuous_batch.rs</scenario> <!-- path to the scenario folder on remote node -->
<delayTime>100</delayTime> <!-- time delayed in milliseconds on nodes so that data can be written and retrieved -->
<removeFiles>true</removeFiles> <!-- remove output files or not from remote nodes after simulation runs--><!-- flag to automatically create a user project jar, using build.xml, to transfer to nodes for each set of scenarios -->
<createJar>false</createJar>
```

FIGURE 15. XML_Launch_Inputs.xml

```
1<model name="PredatorPrey">
2  <classpath>
3    <agents path="/Applications/RepastUserEnvironment11/workspace/repast.simphony.demo.predatorprey/bin"/>
4    <agents path="/Applications/gridgain-2.1.1/bin"/>
5  </classpath>
6</model>
```

FIGURE 16. Remote Bin Folder

```
1<model name="PredatorPrey">
2  <classpath>
3    <agents path="/Applications/RepastUserEnvironment11/workspace/repast.simphony.demo.predatorprey/bin"/>
4    <agents path="/Applications/gridgain-2.1.1/bin/bin"/>
5  </classpath>
6</model>
```

FIGURE 17. The reference to the remote bin folder seen for projects using ant build.xml file. The two /bin references are needed because the user's jar has its own bin folder.

```

GridGain Command Line Loader, ver. 2.1.1
Copyright (C) 2005-2009 GridGain Systems.

[11:45:55,615][INFO ][main][GridKernel\NULL]
[11:45:55,615][INFO ][main][GridKernel\NULL] VM arguments: [-ea, -XX:MaxPermSize=864m, -XX:MaxNewSize=864m, -XX:SurvivorRatio=128, -XX:MaxTenuringThreshold=0, -XX:UseLB, -XX:UseConcurrentSweepGC, -XX:+CMSClassUnloadingEnabled, -Dcom.sun.management.jmxremote, -DGRIDGAIN_HOME=/Applications/gridgain-2.1.1/, -DGRIDGAIN_PROG_NAME=gridgain.sh]
[11:45:55,615][INFO ][main][GridKernel\NULL] GridGain license file can be at: /Applications/gridgain-2.1.1/license.txt
[11:45:55,615][INFO ][main][GridKernel\NULL] 3rd party licenses can be found at: /Applications/gridgain-2.1.1//libs/licenses
[11:45:56,036][INFO ][main][XeJBDefinitionReader] Loading XML bean definitions from resource loaded from byte array
[11:45:56,382][INFO ][main][GridKernel\NULL] User class loader version: 0
[11:45:56,382][INFO ][main][GridKernel\NULL] Local node ID: 30087bf4d-1819-489c-cd27-fcf01099ff7c6
[11:45:56,382][INFO ][main][GridKernel\NULL] P2P excluded path: []
[11:45:56,382][INFO ][main][GridKernel\NULL] Peer class loading disabled: true
[11:45:56,382][INFO ][main][GridKernel\NULL] Peer class loading missed resources cache size: 100
[11:45:56,382][INFO ][main][GridKernel\NULL] Peer class loading timeout (ms): 5000
[11:45:56,383][INFO ][main][GridKernel\NULL] Metrics expiration time: 9223372036854775807
[11:45:56,383][INFO ][main][GridKernel\NULL] Metrics history size: 1000000
[11:45:56,383][INFO ][main][GridKernel\NULL] Metrics stats delete (ms): 0
[11:45:56,401][INFO ][main][GridKernel\NULL] Object marshaller: GridKStreamMarshaller []
[11:45:56,401][INFO ][main][GridKernel\NULL] Grid executor service [name=executorService, corePoolSize=100, maxPoolSize=100, keepAliveTime=0ms, queueCls=LinkedBlockingQueue, threadFactoryCls=GridThreadFactory, rejectionHandlerCls=AbortPolicy]
[11:45:56,401][INFO ][main][GridKernel\NULL] Grid executor service [name=systemExecutorService, corePoolSize=5, maxPoolSize=5, keepAliveTime=0ms, queueCls=LinkedBlockingQueue, threadFactoryCls=GridThreadFactory, rejectionHandlerCls=AbortPolicy]
[11:45:56,401][INFO ][main][GridKernel\NULL] Grid executor service [name=peerClassLoaderingExecutorService, corePoolSize=20, maxPoolSize=20, keepAliveTime=0ms, queueCls=LinkedBlockingQueue, threadFactoryCls=GridThreadFactory, rejectionHandlerCls=AbortPolicy]
[11:45:56,514][INFO ][main][GridLocalMetricsManager] Starting SPI implementation: org.gridgain.grid.spi.metrics.jdk.GridJdkLocalMetricsSpi
[11:45:56,514][INFO ][main][GridLocalMetricsManager] Using parameter [isPreferSigar=true]
[11:45:56,941][INFO ][main][GridTcpCommunicationSpi] Using parameter [sigarCombined=true]
[11:45:56,751][INFO ][main][GridJdkLocalMetricsSpi] SPI started ok [startMs=237, spiName=GridJdkLocalMetricsSpi]
[11:45:56,945][INFO ][main][GridTcpCommunicationSpi] Successfully bound to TCP port: 47108
[11:45:56,945][INFO ][main][GridCommunicationManager] Starting SPI implementation: org.gridgain.grid.spi.communication.tcp.GridTcpCommunicationSpi
[11:45:56,945][INFO ][main][GridTcpCommunicationSpi] Using parameter [localPort=47108]
[11:45:56,945][INFO ][main][GridTcpCommunicationSpi] Using parameter [msgThreads=5]
[11:45:56,945][INFO ][main][GridTcpCommunicationSpi] Using parameter [localPortRange=100]
[11:45:56,945][INFO ][main][GridTcpCommunicationSpi] Using parameter [idleTimeLine=30000]
[11:45:56,945][INFO ][main][GridTcpCommunicationSpi] Using parameter [idleTimeOut=1000]
[11:45:56,945][INFO ][main][GridTcpCommunicationSpi] SPI started ok [startMs=9, spiName=org.gridgain:group=SPIs, name=GridTcpCommunicationSpi]
[11:45:56,963][INFO ][main][GridCheckpointManager] Starting SPI implementation: org.gridgain.grid.spi.checkpoint.sharedfs.GridSharedFsCheckpointSpi
[11:45:56,968][INFO ][main][GridSharedFsCheckpointSpi] Using parameter [folder=/Applications/gridgain-2.1.1/work/checkpoint/sharedfs]
[11:45:56,982][INFO ][main][GridSharedFsCheckpointSpi] SPI started ok [startMs=18, spiName=org.gridgain:group=SPIs, name=GridSharedFsCheckpointSpi]
[11:45:56,991][INFO ][main][GridMemoryEventStorageSpi] Starting SPI implementation: org.gridgain.grid.spi.eventstorage.memory.GridMemoryEventStorageSpi
[11:45:56,991][INFO ][main][GridMemoryEventStorageSpi] Using parameter [expireAfterMs=9223372036854775807]
[11:45:56,991][INFO ][main][GridMemoryEventStorageSpi] Using parameter [expireCnt=100000]

```

FIGURE 18. Launched Remote Node

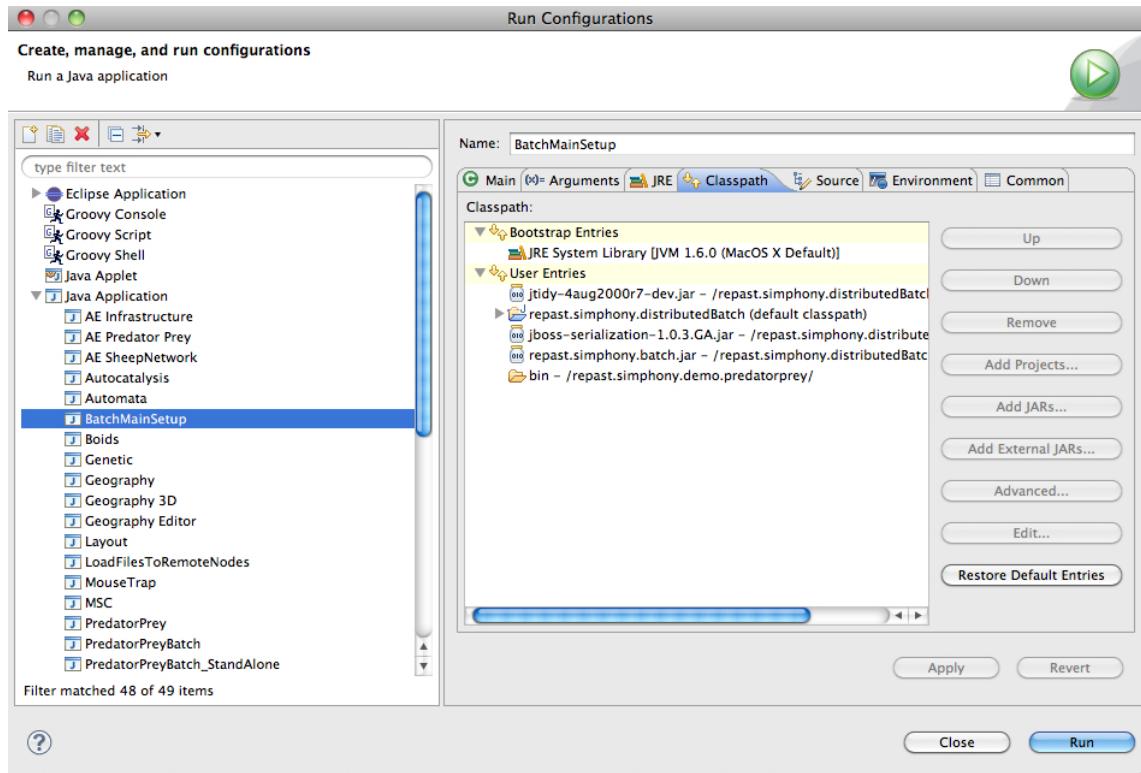


FIGURE 19. Reference To the Local Bin Folder

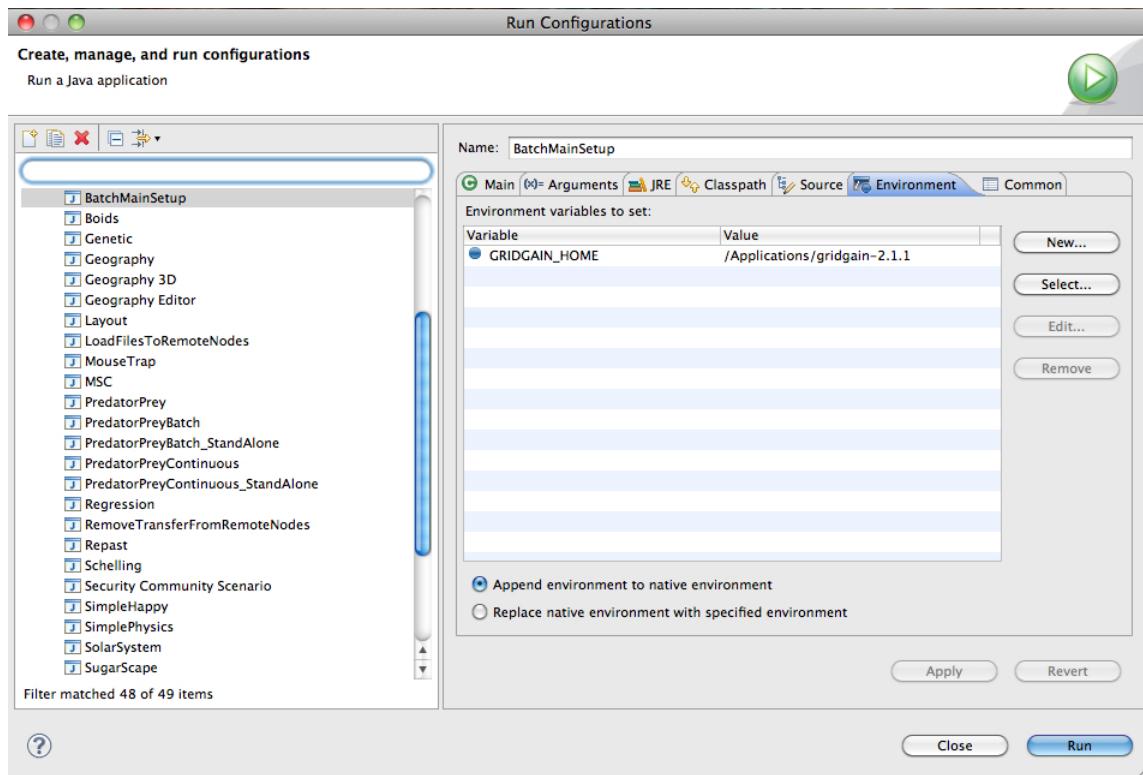


FIGURE 20. Environment reference to the GridGain installation folder.

```
[09:39:38,936][INFO ][main][GridMulticastDiscoverySpi] Waiting for initial heartbeat timeout (3000 milliseconds)
[09:39:41,936][INFO ][main][GridMulticastDiscoverySpi] SPI started ok [startMs=3022, spmBean=org.gridein.grid:group=SPIS, name=gridMulticastDiscoverySpi]
[09:39:42,913][INFO ][main][GridDiscoveryManager]
>>> Discovery Snapshot.
>>>
>>> Number of nodes: 2
>>> Topology hash: 0xC7F76A
>>> Local: FB11A482-6546-460F-951E-4ED4206425AF, 192.229.141.15, Mac OS X x86_64 10.6.4, maitewei, Java(TM) SE Runtime Environment 1.6.0_20-b02-284-10N1134
>>> Remote: F04D3837-299A-43C9-94FC-F448C2A365D4, 192.168.170.107, Mac OS X x86_64 10.6.4, maitewei, Java(TM) SE Runtime Environment 1.6.0_20-b02-279-10M3065
>>> Total number of CPUs: 8

[09:39:42,855][WARN ][main][GridUpdateNotifier] New version is available at www.gridein.org: 3.0.0c-beta.16082010
[09:39:42,876][INFO ][main][GridKernelNull]

>>> -----
>>> GridGain ver. 2.1.1-26022099 STARTED OK in 358ms.
>>>
>>> OS name: Mac OS X 10.6.4 x86_64, 8 CPU(s)
>>> OS user: maitewei
>>> VM information: Java(TM) SE Runtime Environment 1.6.0_20-b02-284-10N1134 Apple Inc. Java HotSpot(TM) 64-Bit Server VM 16.3-b01-284
>>> VM name: 16928Mark-Altwells-MacBook-Pro.local
>>> Optional grid name: null
>>> Local node ID: FB11A482-6546-460F-951E-4ED4206425AF
>>> Local node physical address: 192.229.141.15, en1
>>> GridGain documentation: http://www.gridein.org/product.html

[09:39:43,556][INFO ][grid-uci-scanner][#Knull][GridDeploymentSpi] Found new or updated GAR units [uri=file:///Applications/gridein-2.1.1/work/deployment/file/distributed.gar, file=/var/folders/-6-/6GUHnfoEB8Z88s0Trhu7E++/TI-/Tmp-/gg_deploy/distriuted4170863424634529949, instamp=128433100000]
[09:39:43,755][INFO ][grid-uci-scanner][#Knull][XbeanDefinitionReader] Loading XML bean definitions from resource loaded from byte array
[09:39:43,961][INFO ][grid-uci-scanner][#Knull][GridDeploymentClassloader] Class loader (re)registered [Gridloader-GridDeploymentClassloader [file=/var/folders/-6-/6GUHnfoEB8Z88s0Trhu7E++/TI-/Tmp-/gg_deploy/distriuted4170863424634529949], timestamp=128433100000, url=file:///Applications/gridein-2.1.1/work/deployment/file/distributed.gar, file=/var/folders/-6-/6GUHnfoEB8Z88s0Trhu7E++/TI-/Tmp-/gg_deploy/distriuted4170863424634529949]
[09:39:45,262][INFO ][grid-uci-scanner][#Knull][ScenarioPilotInitializer] Task locally deployed: LocalDeploymentClass [undeployed=false, usage=0, super=GridDeploymentClass [cls=class org.gridgain.grid.uci.GridDeploymentClass, depMode=ISOLATED, glsId=sun.misc.Unsafe$AppClassLoader@ed9dbd, glsId=java.util.concurrent.ConcurrentHashMap@f4f629b, 88664667-98e1-9b5927-86d41d, userVer=0, agnName=1, agnLast=nearest, simphony, batch], distributed=gridgain.GridBatchTask, local=true]
[09:39:46,356][WARN ][grid-uci-scanner][#Knull][ScenarioPilotInitializer] Warning while parsing classpath: Empty or non-existent path '/Applications/gridein-2.1.1/bin'.
[09:39:46,356][WARN ][grid-uci-scanner][#Knull][ScenarioPilotInitializer] Warning while parsing classpath: Empty or non-existent path '/Applications/gridein-2.1.1/bin'.
[09:39:46,448][WARN ][grid-uci-scanner][#Knull][ScenarioPilotWatchesInitializer] While initializing watches: Empty or non-existent path '/Applications/gridein-2.1.1/bin'.
[09:39:46,448][WARN ][grid-uci-scanner][#Knull][ScenarioPilotWatchesInitializer] While initializing watches: Empty or non-existent path '/Applications/gridein-2.1.1/bin'.
[09:39:47,235][WARN ][grid-uci-scanner][#Knull][ScenarioPilotWatchesInitializer] While initializing watches: Empty or non-existent path '/Applications/gridein-2.1.1/bin'.
[09:39:47,235][WARN ][grid-uci-scanner][#Knull][ScenarioPilotWatchesInitializer] While initializing watches: Empty or non-existent path '/Applications/gridein-2.1.1/bin'.
[09:39:52,277][WARN ][DeploymentLevel][Thread-3][{}][Agent Count]: [Tick=1.0, wolf_count=51.0, GRASPOLI:simphony,data.log.message.size=0, Run Number=31]
[09:39:52,346][WARN ][DeploymentLevel][Thread-3][{}][Agent Count]: [Tick=6.0, wolf_count=58.0, GRASPOLI:simphony,data.log.message.size=0, Run Number=31]
[09:39:52,398][WARN ][DeploymentLevel][Thread-3][{}][Agent Count]: [Tick=11.0, wolf_count=58.0, GRASPOLI:simphony,data.log.message.size=0, Run Number=31]
[09:40-52,430][WARN ][DeploymentLevel][Thread-3][{}][Agent Count]: [Tick=16.0, wolf_count=58.0, GRASPOLI:simphony,data.log.message.size=0, Run Number=31]
```

FIGURE 21. Local node distributed batch output.

```
>>> !recoverory snapshot
>>> Number of nodes: 1
>>> Topology hash: 0xCD1BB5D8
>>> Local: 54DD3B37-299A-43C9-94FC-F448CB2A36D4, 192.168.170.107, Mac OS X x86_64 10.6.4, mallowee, Java(TM) SE Runtime Environment 1.6.0_20-b02-279-10M3065
>>> Total number of CPUs: 4

[09:37:51.293][INFO] Ignite[grid=102%null%][GridDeploymentPerLoaderStore] Task was undeployed in Private or Isolated mode: GridDeploymentClass [cls=class repast:symphony:batch:distributed:gridgain.GridGainBatchTask, depMode=ISOLATED, olsId=GridDeploymentClass.Loader[singleNode=true, nodeLdrMap={3d4953-2cc2-4ad8-91ab-37fe1c6a3e9e=GridPair [val1=dac4f27.3a4f-4da3-9aaa:c2765982eb1, val2=1]}, p2pTimeout=5000], olsLdr=dac4f27.3a4f-4da3-9aaa:c2765982eb1, userVer=0, seqNum=1, alias=repast:symphony:batch:distributed:gridgain.GridGainBatchTask, local=false]
[09:39:31.521][INFO] Ignite[grid=102%null%][GridDiscoveryManager] Added new node to topology: GridMulticastDiscoveryNode [id=b11a482-65a6-460f-951e-4ed4206425af, state=READY, lastHeartbeat=128457378940, addr=192.229.141.15, port=47301, startTime=128457378941]
[09:39:33.259][INFO] Ignite[grid=102%null%][GridDiscoveryManager]

>>>
>>> Discovery Snapshot.
>>> Number of nodes: 2
>>> Topology hash: 0x7F76A
>>> Local: 54DD3B37-299A-43C9-94FC-F448CB2A36D4, 192.168.170.107, Mac OS X x86_64 10.6.4, mallowee, Java(TM) SE Runtime Environment 1.6.0_20-b02-279-10M3065
>>> Remote: FB11A482-65A6-460F-951E-4ED4206425AF, 192.229.141.13, Mac OS X x86_64 10.6.4, mallowee, Java(TM) SE Runtime Environment 1.6.0_20-b02-284-10N1134
>>> Total number of CPUs: 8

[09:39:45.533][INFO] Ignite[grid=102%null%][GridDeploymentPerLoaderStore] Task was deployed in Private or Isolated mode: GridDeploymentClass [cls=class repast:symphony:batch:distributed:gridgain.GridGainBatchTask, depMode=ISOLATED, olsId=GridDeploymentClass.Loader[singleNode=true, nodeLdrMap={011a32-856e-4691-9057-1e4d4206425af=GridPair [val1=c1f694b-8866-4d7r-9ba1-90597d864101, val2=1]}, p2pTimeout=5000], olsLdr=d1f4694b-8866-4d7r-9ba1-90597d864101, userVer=0, seqNum=1, alias=repast:symphony:batch:distributed:gridgain.GridGainBatchTask, local=false]
/Applications/repastUserEnvironment/1/workspace/repast:symphony:demo:predator/prey/breeder_prey, continuous, batch.rs
[09:39:49.651][INFO] Repast[evl=Thread-50][0] [Agent Count: {Tick=1.0, wolf=50, repast:symphony:datalog:message:status=NONE, sheep=96.0, Run Number=0}]
[09:39:49.663][INFO] Repast[evl=Thread-50][0] [Agent Count: {Tick=6.0, wolf=50, repast:symphony:datalog:message:status=NONE, sheep=96.0, Run Number=0}]
[09:39:49.675][INFO] Repast[evl=Thread-50][0] [Agent Count: {Tick=11.0, wolf=50, repast:symphony:datalog:message:status=NONE, sheep=96.0, Run Number=0}]
[09:39:49.677][INFO] Repast[evl=Thread-50][0] [Agent Count: {Tick=16.0, wolf=50, repast:symphony:datalog:message:status=NONE, sheep=96.0, Run Number=0}]
[09:39:49.679][INFO] Repast[evl=Thread-50][0] [Agent Count: {Tick=21.0, wolf=50, repast:symphony:datalog:message:status=NONE, sheep=96.0, Run Number=0}]
[09:39:49.681][INFO] Repast[evl=Thread-50][0] [Agent Count: {Tick=26.0, wolf=50, repast:symphony:datalog:message:status=NONE, sheep=96.0, Run Number=0}]
[09:39:49.683][INFO] Repast[evl=Thread-50][0] [Agent Count: {Tick=31.0, wolf=50, repast:symphony:datalog:message:status=NONE, sheep=96.0, Run Number=0}]
[09:39:50.115][INFO] Repast[evl=Thread-50][0] [Agent Count: {Tick=36.0, wolf=50, repast:symphony:datalog:message:status=NONE, sheep=96.0, Run Number=0}]
[09:39:50.117][INFO] Repast[evl=Thread-50][0] [Agent Count: {Tick=41.0, wolf=50, repast:symphony:datalog:message:status=NONE, sheep=96.0, Run Number=0}]
[09:39:50.119][INFO] Repast[evl=Thread-50][0] [Agent Count: {Tick=46.0, wolf=50, repast:symphony:datalog:message:status=NONE, sheep=96.0, Run Number=0}]
[09:39:51.089][INFO] Repast[evl=Thread-50][0] [Agent Count: {Tick=51.0, wolf=50, repast:symphony:datalog:message:status=NONE, sheep=96.0, Run Number=0}]
[09:39:51.337][INFO] Repast[evl=Thread-50][0] [Agent Count: {Tick=56.0, wolf=50, repast:symphony:datalog:message:status=NONE, sheep=96.0, Run Number=0}]
[09:39:51.351][INFO] Repast[evl=Thread-50][0] [Agent Count: {Tick=61.0, wolf=50, repast:symphony:datalog:message:status=NONE, sheep=96.0, Run Number=0}]
[09:39:53.111][INFO] Repast[evl=Thread-50][0] [Agent Count: {Tick=67.0, wolf=50, repast:symphony:datalog:message:status=NONE, sheep=96.0, Run Number=0}]
[09:39:53.113][INFO] Repast[evl=Thread-50][0] [Agent Count: {Tick=71.0, wolf=50, repast:symphony:datalog:message:status=NONE, sheep=96.0, Run Number=0}]
[09:39:53.115][INFO] Repast[evl=Thread-50][0] [Agent Count: {Tick=76.0, wolf=50, repast:symphony:datalog:message:status=NONE, sheep=96.0, Run Number=0}]
[09:39:53.183][INFO] Repast[evl=Thread-50][0] [Agent Count: {Tick=81.0, wolf=50, repast:symphony:datalog:message:status=NONE, sheep=96.0, Run Number=0}]
[09:39:52.053][INFO] Repast[evl=Thread-50][0] [Agent Count: {Tick=86.0, wolf=50, repast:symphony:datalog:message:status=NONE, sheep=96.0, Run Number=0}]
[09:39:52.071][INFO] Repast[evl=Thread-50][0] [Agent Count: {Tick=91.0, wolf=50, repast:symphony:datalog:message:status=NONE, sheep=96.0, Run Number=0}]
[09:39:52.073][INFO] Repast[evl=Thread-50][0] [Agent Count: {Tick=96.0, wolf=50, repast:symphony:datalog:message:status=NONE, sheep=96.0, Run Number=0}]
[09:39:22.216][INFO] Repast[evl=Thread-50][0] [Agent Count: {Tick=101.0, wolf=50, repast:symphony:datalog:message:status=NONE, sheep=96.0, Run Number=0}]
[09:39:22.231][INFO] Repast[evl=Thread-50][0] [Agent Count: {Tick=106.0, wolf=50, repast:symphony:datalog:message:status=NONE, sheep=96.0, Run Number=0}]
[09:39:22.233][INFO] Repast[evl=Thread-50][0] [Agent Count: {Tick=111.0, wolf=50, repast:symphony:datalog:message:status=NONE, sheep=96.0, Run Number=0}]
[09:39:22.235][INFO] Repast[evl=Thread-50][0] [Agent Count: {Tick=116.0, wolf=50, repast:symphony:datalog:message:status=NONE, sheep=96.0, Run Number=0}]
[09:39:22.237][INFO] Repast[evl=Thread-50][0] [Agent Count: {Tick=121.0, wolf=50, repast:symphony:datalog:message:status=NONE, sheep=96.0, Run Number=0}]
[09:39:22.249][INFO] Repast[evl=Thread-50][0] [Agent Count: {Tick=126.0, wolf=50, repast:symphony:datalog:message:status=NONE, sheep=96.0, Run Number=0}]
```

FIGURE 22. Remote node distributed batch output.

3. OUTPUTS PRODUCED

Outputs produced by the remote and local nodes will be collected and placed in the /output folder of the users' project (e.g., see the /output folder in repast.simphony.demo.predatorprey). The output files should be unique for each run. Remote nodes' outputs that are transferred to the /output folder may require that the user set a value for the "delayTime" parameter in the XML_Launch_Inputs.xml file to enable output results to be transferred to the local node (i.e., the delay would allow enough time to create the output file and to be transferred over). The local node may also write local node results to the distributedBatch plugin path. The user, if desired, can remove these outputs as they will also be automatically copied over to the /output folder in their project folder. In addition, remote nodes will create output files in the /bin directories of the GridGain installation on remote nodes. These files are automatically removed if the user sets the "removeFiles" option in XML_Launch_Inputs.xml to "true". If the user chooses to remove remote output files, these files will still be transferred to the /output folder in the user's project.